Math 116: Problem Set 6

Owen Jones

2/27/2024

1. (a) If $\gcd(e, 24) = 1$, then $\gcd(e, 3) = 1$ and e is odd.
   By Fermat's Little Theorem, $e^2 \equiv 1 \pmod 3$
   $\begin{cases} e^2 \equiv 16m^2 + 24m + 9 \equiv 1 \pmod 8 & \text{if } e \equiv 3 \pmod 4 \\ e^2 \equiv 16m^2 + 8m + 1 \equiv 1 \pmod 8 & \text{if } e \equiv 1 \pmod 4 \end{cases}$
   Because $e^2 = 1 \pmod{24}$ satisfies the system of conguences

   $$e^2 \equiv 1 \pmod 3$$
   $$e^2 \equiv 1 \pmod 8$$

   the CRT states $e^2 \equiv 1 \pmod{24}$ must be the unique solution to the system.

   (b) $\phi(35) = \phi(5) \cdot \phi(7) = 24$. Thus, $ed \equiv 1 \pmod{24}$. However, we know from part (a) that if e and 24 are coprime, $e^2 \equiv 1 \pmod{24}$.
   Thus, $c^e \equiv (m^e)^e \equiv m^{e^2} \equiv m^{\phi(35)k} \cdot m \equiv m \pmod{35}$

2. $\gcd(e, (p-1)(q-1)(r-1)) = 1$ and $\gcd(d, (p-1)(q-1)(r-1)) = 1$.

3. No. It is equivalent to using a single encryption exponent $e^* = e_1 \cdot e_2$. It is not any more difficult to find a d s.t $de^* \equiv 1 \pmod{\phi(n)}$ i.e it still only depends on how difficult it is to factor n.

4. From the information given $n \mid (516107 \cdot 187722 - 14)(516107 \cdot 187722 + 14)$. It follows $\gcd(n, 516107 \cdot 187722 - 14) = 1129$ which is a non-trivial factor of n, the other being 569.

   $$(516107 \cdot 187722 - 14) = 642401 \cdot 150816 + 289024$$
   $$642401 = 2 \cdot 289024 + 64353$$
   $$289024 = 4 \cdot 64353 + 31612$$
   $$64353 = 2 \cdot 31612 = 1129$$
   $$31612 = 28 \cdot 1129$$

5. $m_B - m_A \equiv p \cdot p^{-1} \pmod n$ by the CRT where $p \cdot p^{-1} \equiv 1 \pmod q$. $0 < p \cdot p^{-1} < n$ because $0 < p^{-1} < q$. It follows $\gcd(p \cdot p^{-1}, n) = p$ gives one of the non-trivial factors of n.

6. (a) If α is a primitive root, then $\alpha^{L_\alpha(\beta_1 \cdot \beta_2)} \equiv \alpha^{L_\alpha(\beta_1)+L_\alpha(\beta_2)} \pmod{p}$ iff $L_\alpha(\beta_1 \cdot \beta_2) \equiv L_\alpha(\beta_1) + L_\alpha(\beta_2) \pmod{p-1}$. Because α is a primitive root, L_α is onto. $\alpha^{L_\alpha(\beta_1 \cdot \beta_2)} \equiv \beta_1 \cdot \beta_2 \equiv \alpha^{L_\alpha(\beta_1)} \cdot \alpha^{L_\alpha(\beta_2)} \equiv \alpha^{L_\alpha(\beta_1)+L_\alpha(\beta_2)} \pmod{p}$.

   (b) Since α is not necesarily a primitive root, $k \leq p-1$ where k is the smallest integer s.t $\alpha^k \equiv 1 \pmod{p}$. Let $x = L_\alpha(\beta_1 \cdot \beta_2), y = L_\alpha(\beta_1)$, and $z = L_\alpha(\beta_2)$ for $0 \leq x, y, z < k$. It follows $\alpha^x \equiv \beta_1 \cdot \beta_2 \equiv \alpha^y \cdot \alpha^z \pmod{p}$. Because $a^{x-y-z} \equiv 1 \equiv \alpha^k \pmod{p} \Rightarrow x \equiv y + z \pmod{k}$. Thus, $L_\alpha(\beta_1 \cdot \beta_2) \equiv L_\alpha(\beta_1) + L_\alpha(\beta_2) \pmod{k}$.

7. (a) $L_2(24) \equiv 3L_2(2) + L_2(3) \pmod{100}$. $L_2(2) = 1 \Leftrightarrow 2^1 \equiv 2 \pmod{101}$ trivially. Thus, $L_2(24) \equiv 72 \pmod{100}$.

   (b) Given $5^3 \equiv 24 \pmod{101}$, it follows $2^{L_2(24)} \equiv 2^{3L_2(5)} \pmod{101}$. Thus, $L_2(24) \equiv 3L_2(5) \equiv 3 \cdot 24 \equiv 72 \pmod{100}$.

8. Given $3^6 \equiv 44 \pmod{137}$ and $3^{10} \equiv 2 \pmod{137}$, it follows $L_3(44) = 6$ and $L_3(2) = 10$ $L_3(44) - 2L_3(2) \equiv L_3(11) \equiv -14 \pmod{136}$. Thus, $L_3(11) \equiv 122 \pmod{136}$.

9. (a) Discrete logarithms are an example of a one way function. Computing $b^y \pmod{p}$ to check password against the list of encrypted passwords is computationally easy. However, given the list of encrypted passwords, it is computationally difficult to deduce the password, x, from $b^x \pmod{p}$ because checking every x between 0 and $p-1$ would take centuries to compute when p is of the order of magnitude 10^{499}.

   (b) The system in part (a) would not be secure if p was a 5 digit number because an exhaustive search of every x between 0 and $p-1$ is feasible with a sufficiently fast computer. Thus, this system would be weak to a brute force attack.

10. If $r = 7 \Rightarrow r^{-1} \equiv 5 \pmod{17}$.
    Thus, $(r^{-1})^a \beta^k m \equiv m \equiv 5^6 \cdot 6 \equiv 12 \pmod{17}$.
    Hence, $m = 12$.

11. (a) If $0 \leq m < n_i$ for $i = 1, 2, 3$, $0 \leq m^2 < mn_1 < n_1 n_2$
    $\Rightarrow 0 \leq m^3 < m^2 n_3 < n_1 n_2 n_3$.
    Thus, $0 \leq m^3 < n_1 n_2 n_3$.

    (b) Let $N = n_1 n_2 n_3$, $z_i = \frac{N}{n_i}$, $y_i \equiv z_i^{-1} \pmod{n_i}$
    Thus, $c_1 y_1 z_1 + c_2 y_2 z_2 + c_3 y_3 z_3$ satisfies the system of conguences.
    Let $m^3 \equiv c_1 y_1 z_1 + c_2 y_2 z_2 + c_3 y_3 z_3 \pmod{n_1 n_2 n_3}$ be the smallest positive integer that satisfies the congruence relation.

    (c) $m = 2305201821192020180514 20$ WETRUSTTRENT
    Found using bisection method.

12. (a) $p = 3994211774931437561721507289$
    $q = 7718138030199014069125222 67$

(b) $m = 805250221040425$ HEYBUDDY

13. $3^{1234} \equiv 8576 \pmod{53047}$

14. (a) $2^{2000} \equiv 3925 \pmod{3989}, 2^{3000} \equiv 1046 \pmod{3989}$

    (b) $L_2(3925 \cdot 1046) \equiv L_2(3925) + L_2(1046) \equiv 5000 \equiv 1012 \pmod{3988}$.
    Thus, $L_2(3925 \cdot 1046) = 1012$.

In [93]:
```python
import numpy as np
import math116
import scipy
from scipy import optimize
import math
```

In [94]:
```python
n_1=1067630413187841523694537298073305552274776079802902672351039
n_2=741591202370072789953706745485666075004784174022368180242037
n_3=667336142291948937637980407048251181747364391891428340555141
N=n_1*n_2*n_3
```

In [95]:
```python
c_1=529845560668797629400939585461719431833561498816920423702247
c_2=169291735293877329351269953081439652585988812455417922505176
c_3=642418962414073836488116737694096521023718712673159264182195
```

In [96]:
```python
y_1=N//n_1
y_2=N//n_2
y_3=N//n_3
```

In [97]:
```python
z_1=math116.inverse(y_1,n_1)
z_2=math116.inverse(y_2,n_2)
z_3=math116.inverse(y_3,n_3)
```

In [98]:
```python
m_3=(c_1*y_1*z_1+c_2*y_2*z_2+c_3*y_3*z_3)%N
```

In [84]:
```python
f = lambda x: x**3-m_3
```

In [99]:
```python
m_3
```

Out[99]:
```
12249739749784771985364504924805398662123078918189011371891240923288000
```

In [110…
```python
def bisection(f,a,b,tol=1):
    if np.sign(f(a))==np.sign(f(b)):
        print('a and b do not bound a root')
    m=(a+b)//2
    if abs(f(m))<tol:
        return m
    elif np.sign(f(a))==np.sign(f(m)):
        return bisection(f,m,b,tol=1)
    elif np.sign(f(b))==np.sign(f(m)):
        return bisection(f,a,m,tol=1)
```

In [111…
```python
bisection(f,m_0,m_3)
```

Out[111…
```
230520182119202018051420
```

In [112…

```
math116.num_to_text(2305201821192020180514 20)
```

Out[112… 'WETRUSTTRENT'

In [102…

```
m=int(pow(m_3,1/3))
```

Out[102… 2.305201821192013e+23

In [33]:

```
a=2
i=2
n=3082787780076703322597022112433309015881410588015304163
while True:
    a=pow(a,i,n)
    p=math116.gcd(a-1,n)
    if p>1:
        print(p)
        break
    i+=1
```

399421177493143756172150728 9

In [34]:

```
q=n//p
```

In [35]:

```
phi_n=(p-1)*(q-1)
```

In [36]:

```
d=math116.inverse(65537,phi_n)
```

In [39]:

```
c=1409434396818034663404225667133198898377678131865927114
pow(c,d,n)
```

Out[39]: 805250221040425

In [46]:

```
pow(3,1234,53047)
```

Out[46]: 8576

In [48]:

```
pow(2,2000,3989)
```

Out[48]: 3925

In [50]:

```
pow(2,3000,3989)
```

Out[50]: 1046

In [51]:

```
pow(2,1012,3989)
```

Out[51]: 869

In [52]:
```
(3925*1046)%3989
```

Out[52]: 869

In [17]:
```
math116.num_to_text(805250221040425)
```

Out[17]: 'HEYBUDDY'

In [ ]:
```

```